

FIG 1

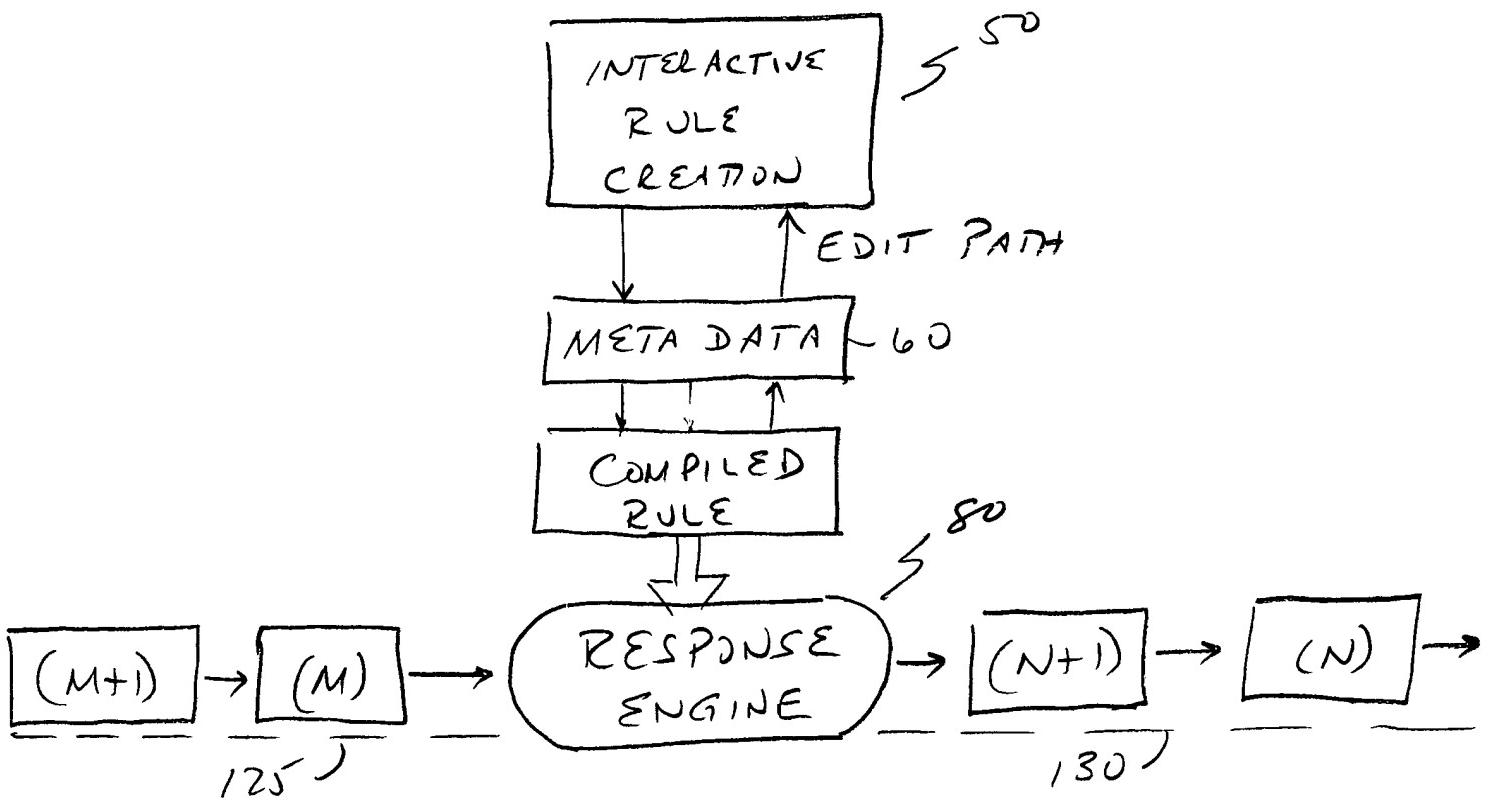


FIG 2

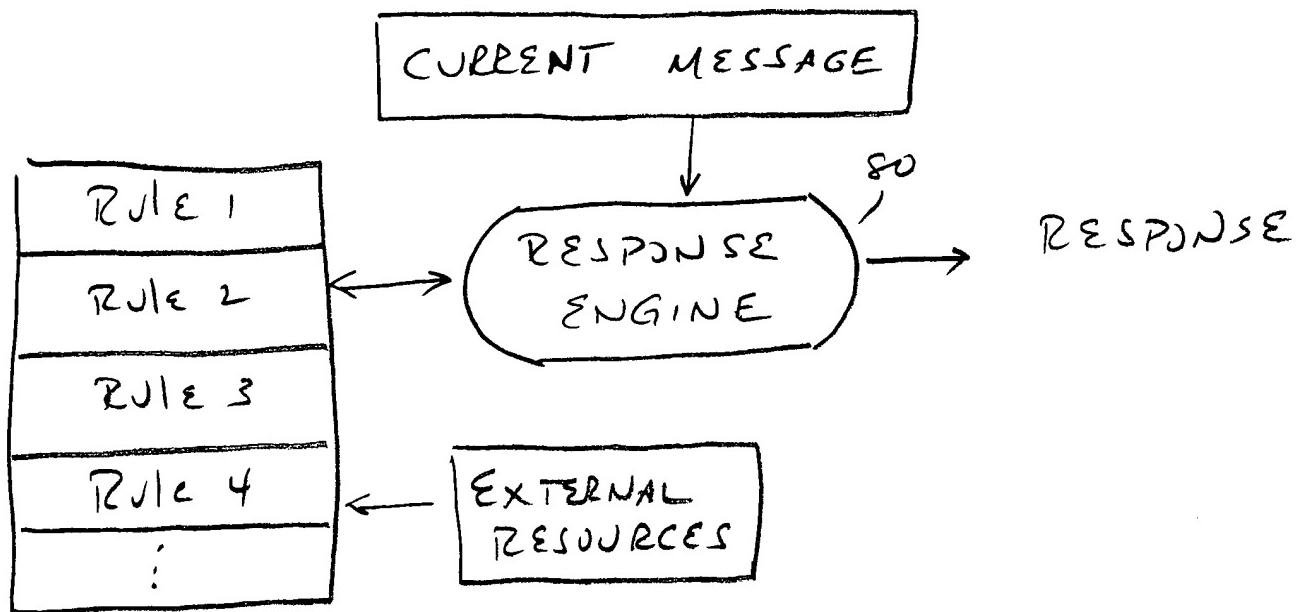


FIG 3

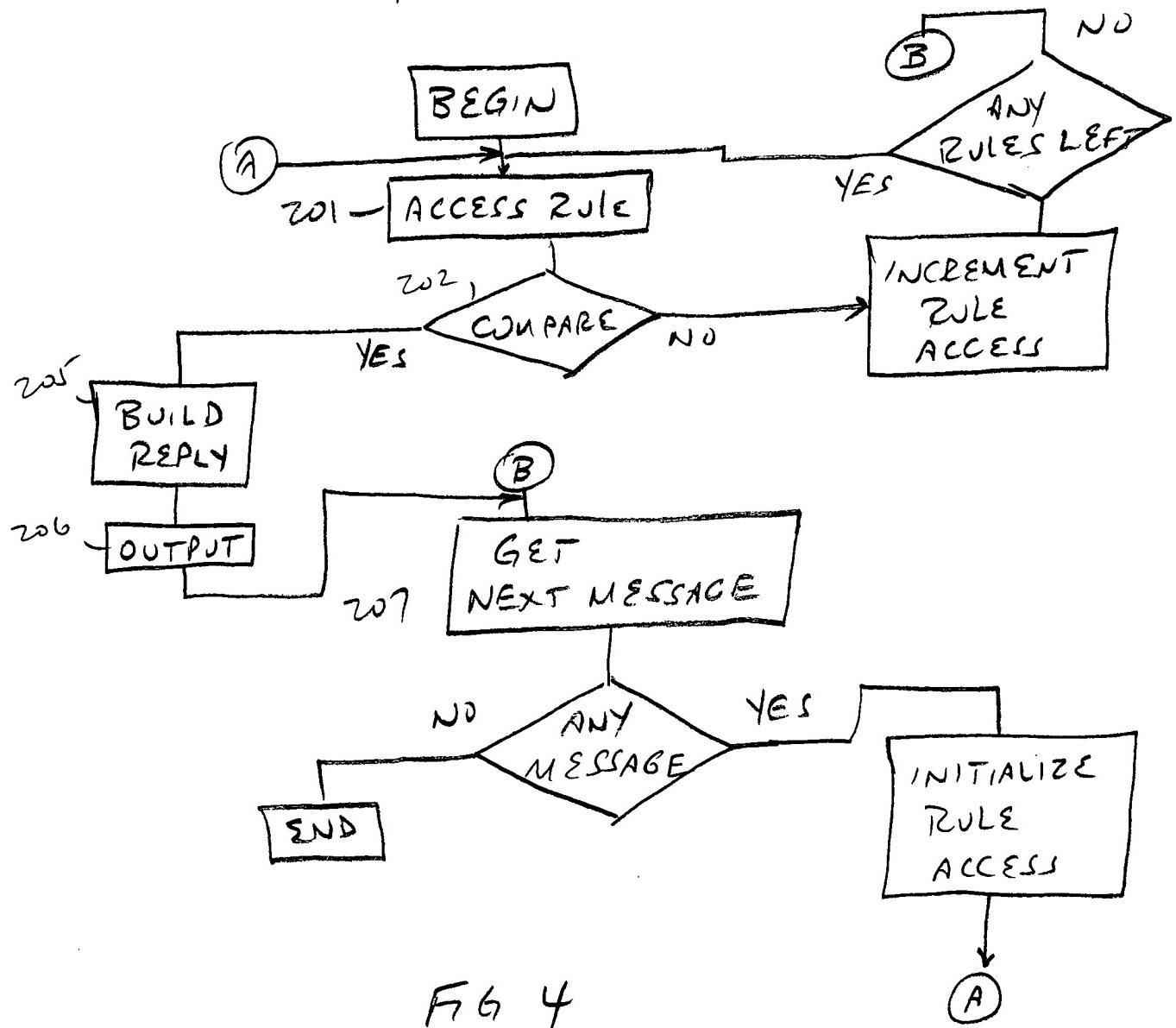
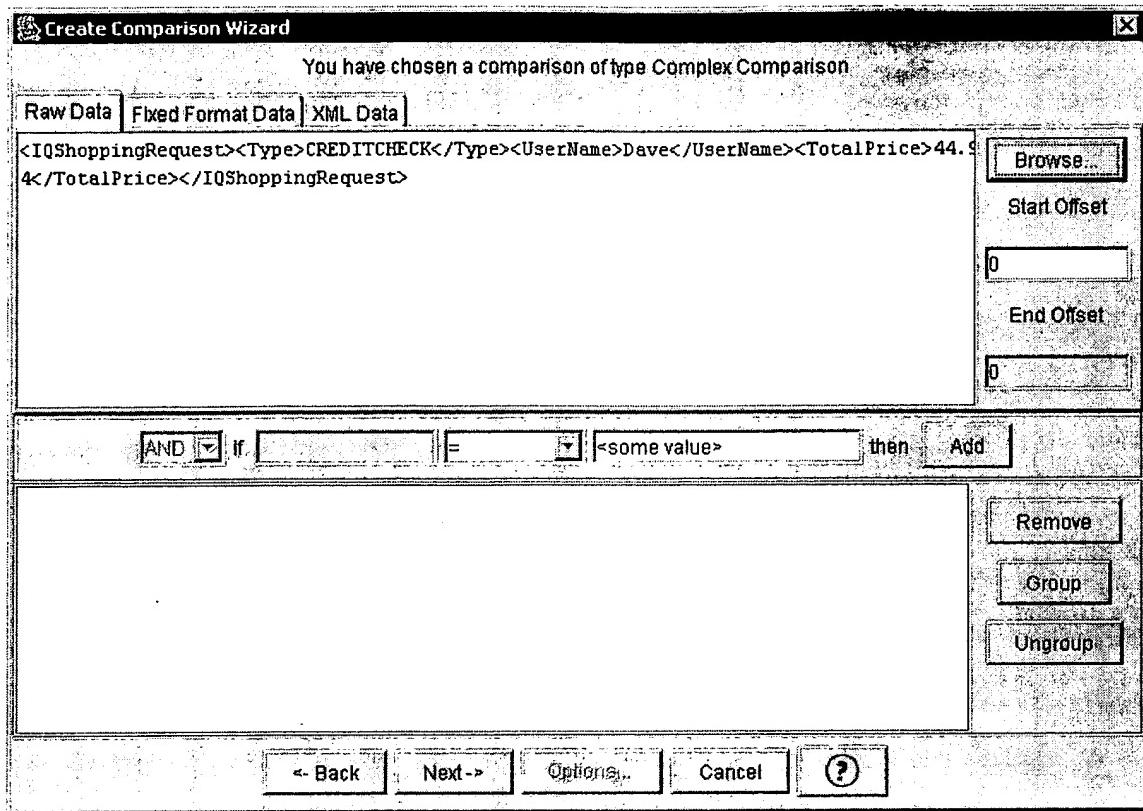
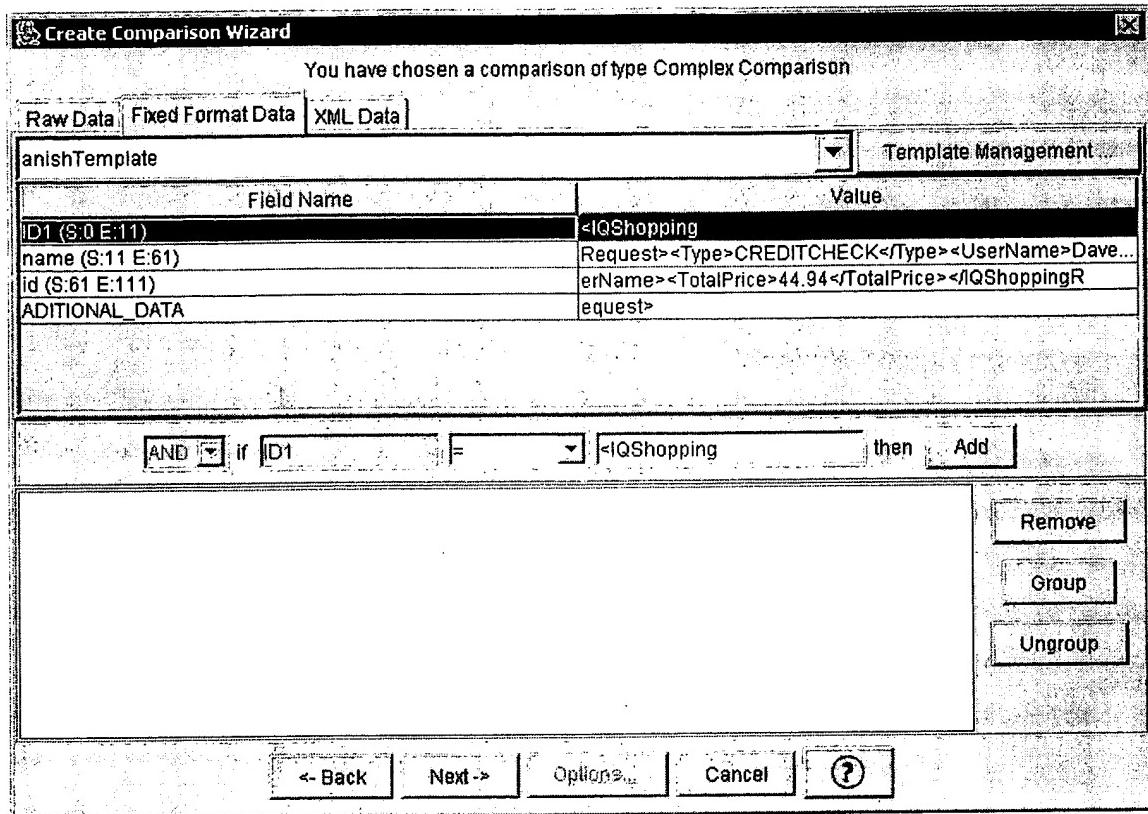


FIG 4



FG ↴



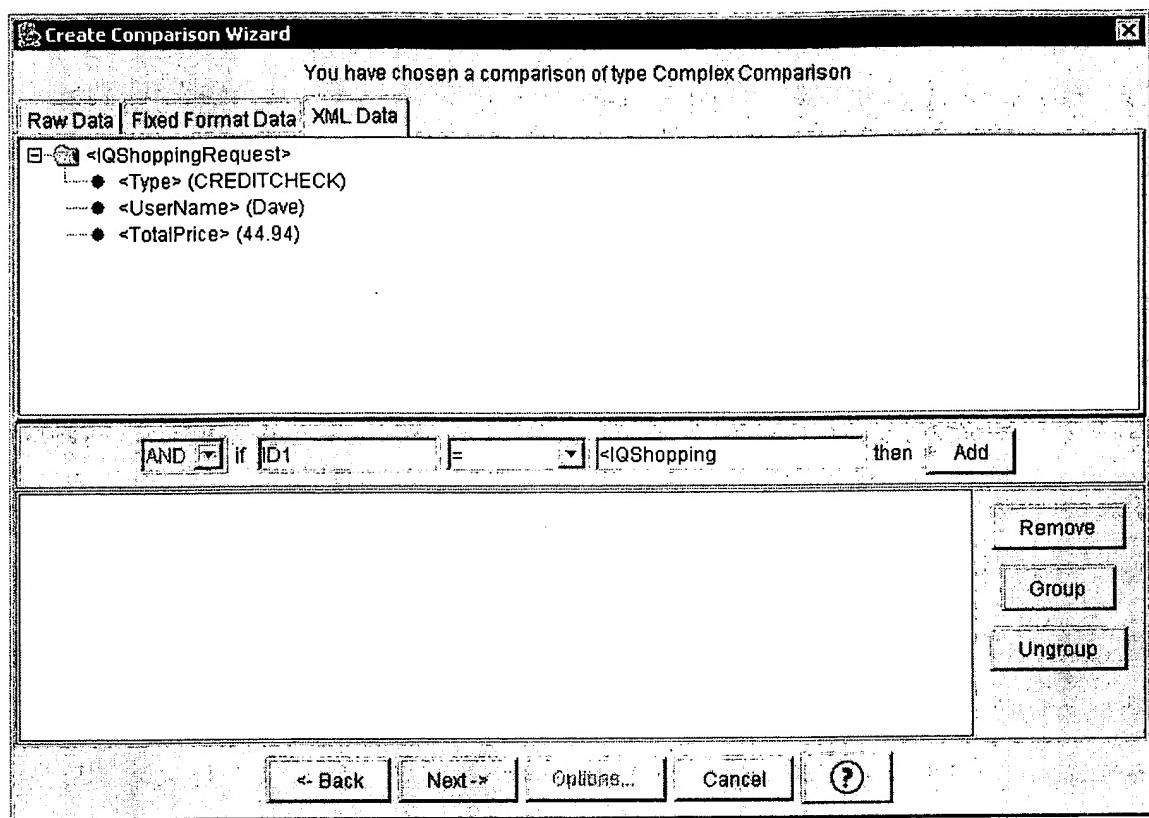
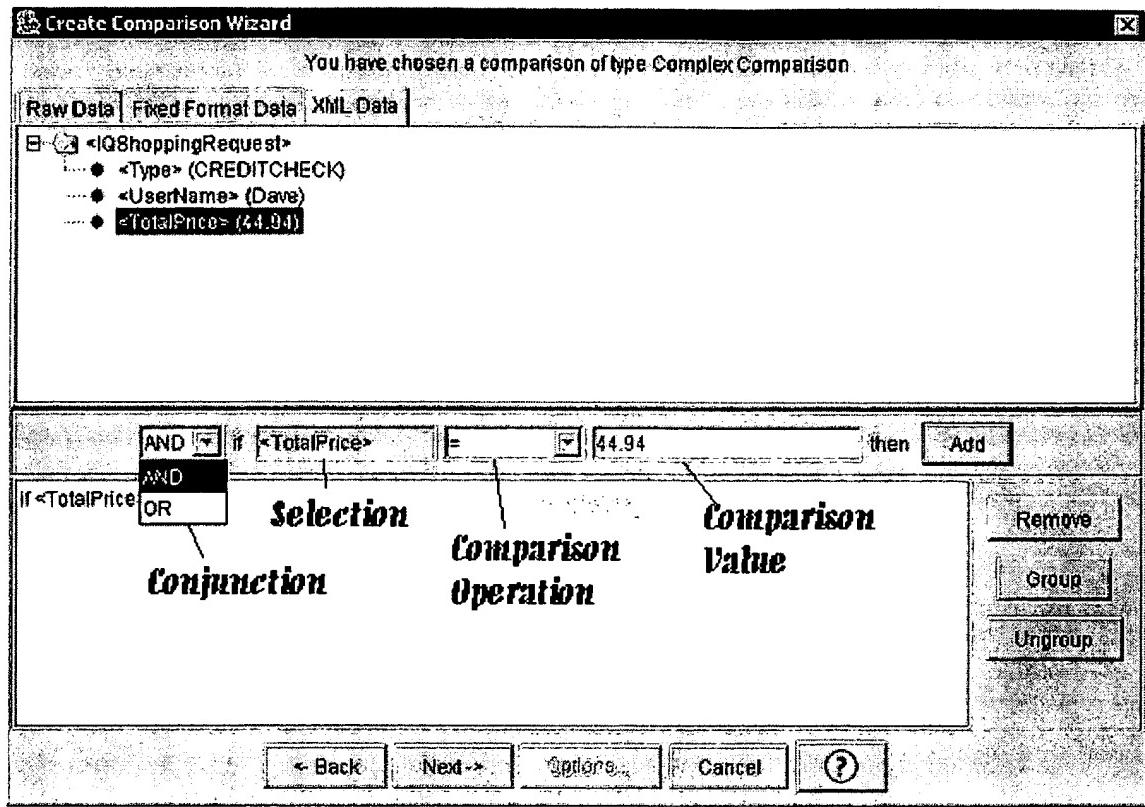


FIG 7



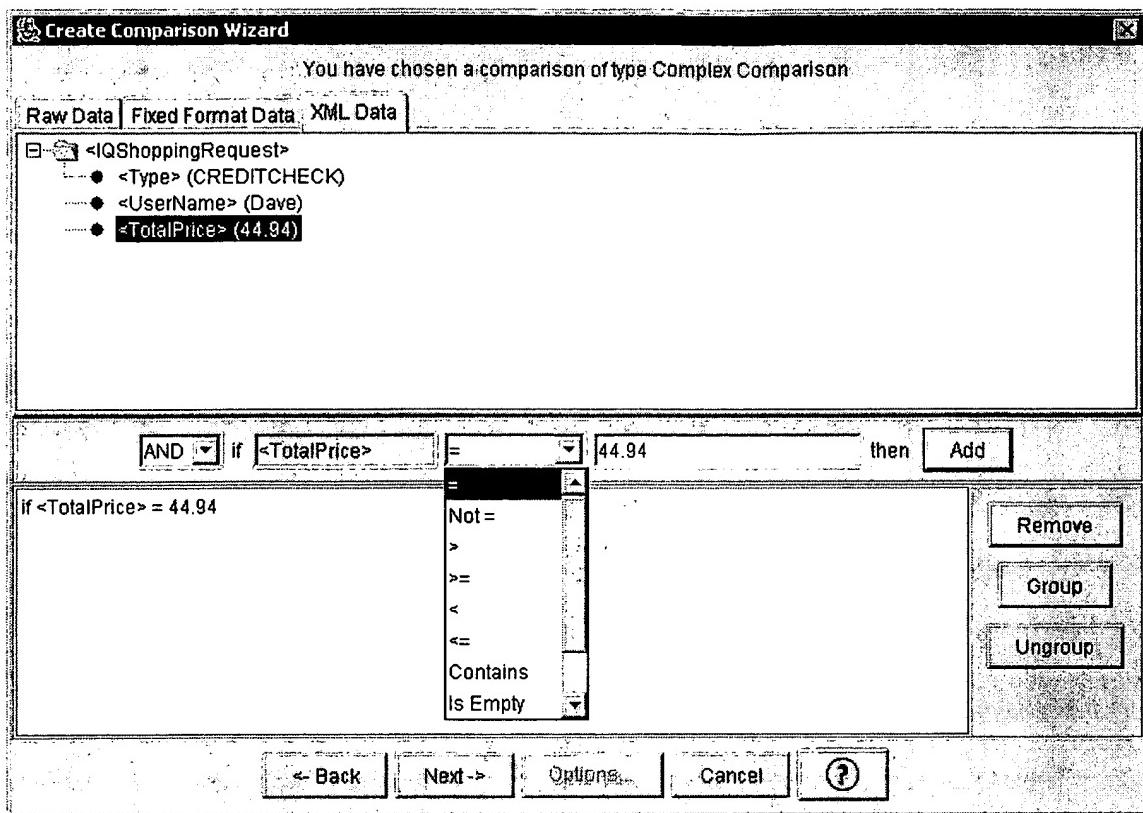
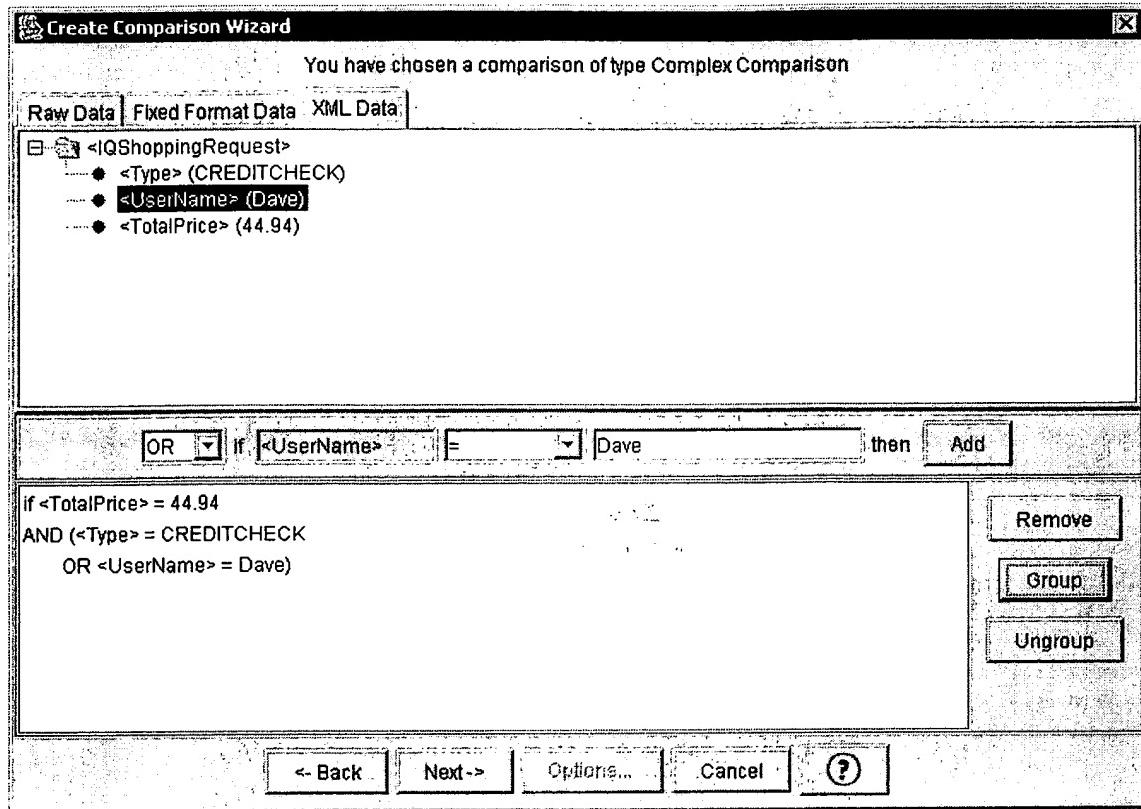
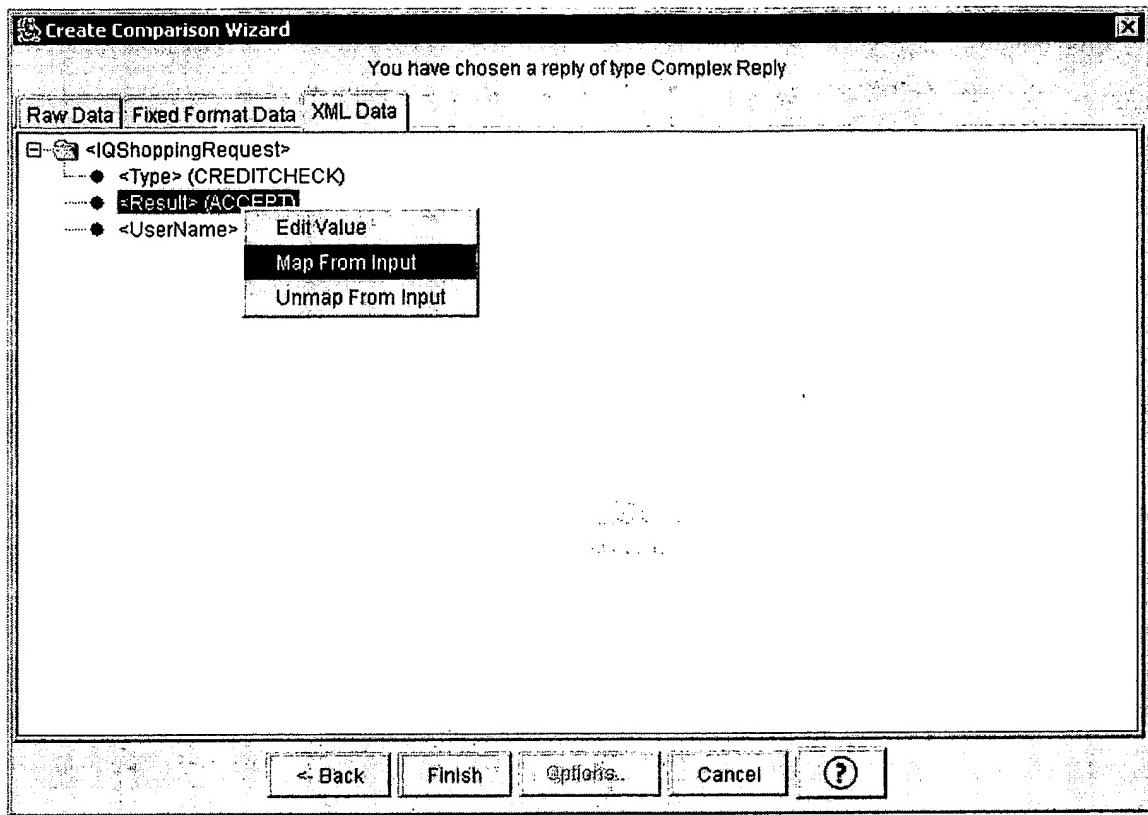


FIG 9



FG 10



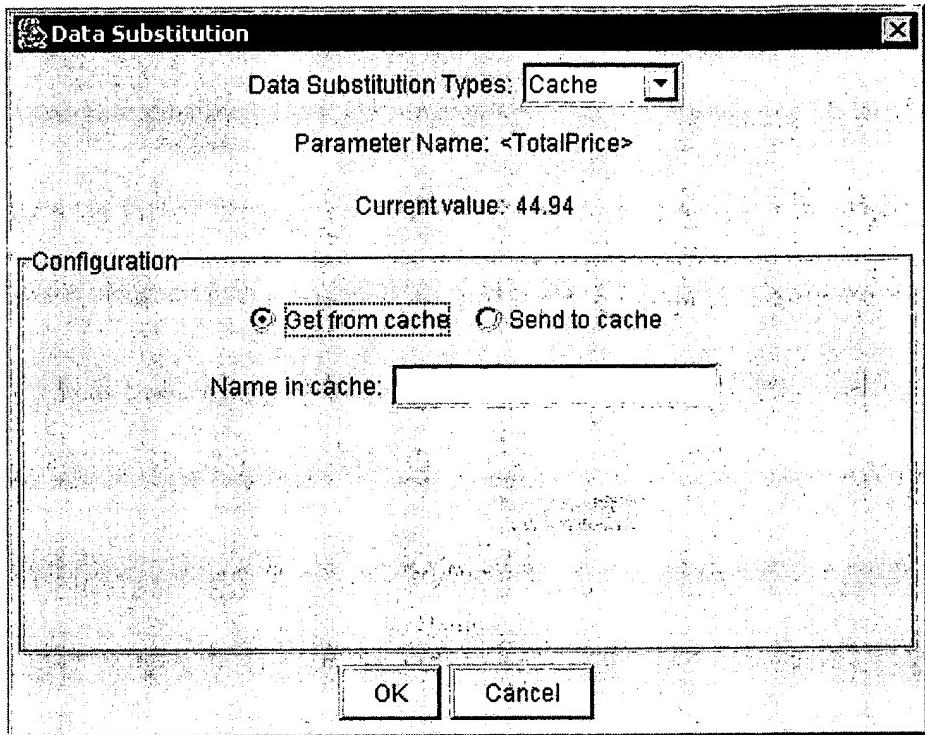


Fig 12

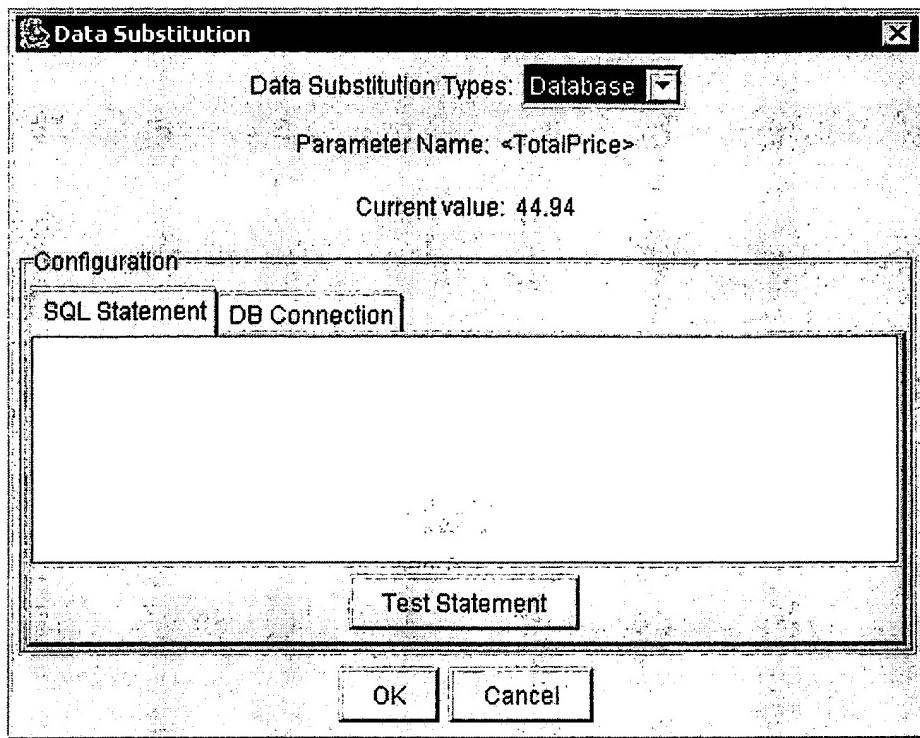


Fig 13

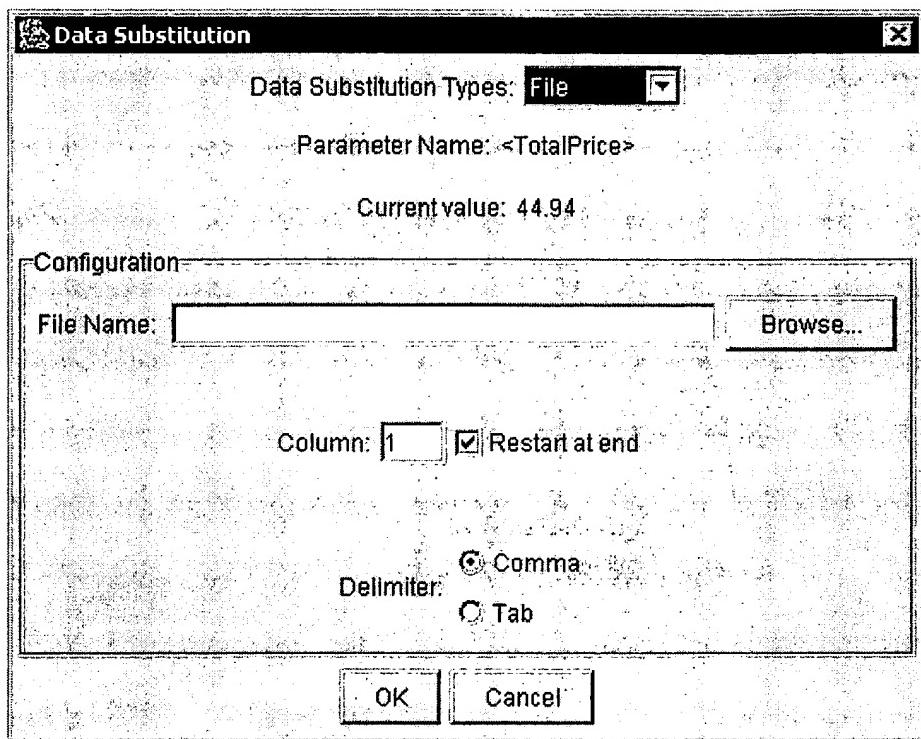


Fig 14

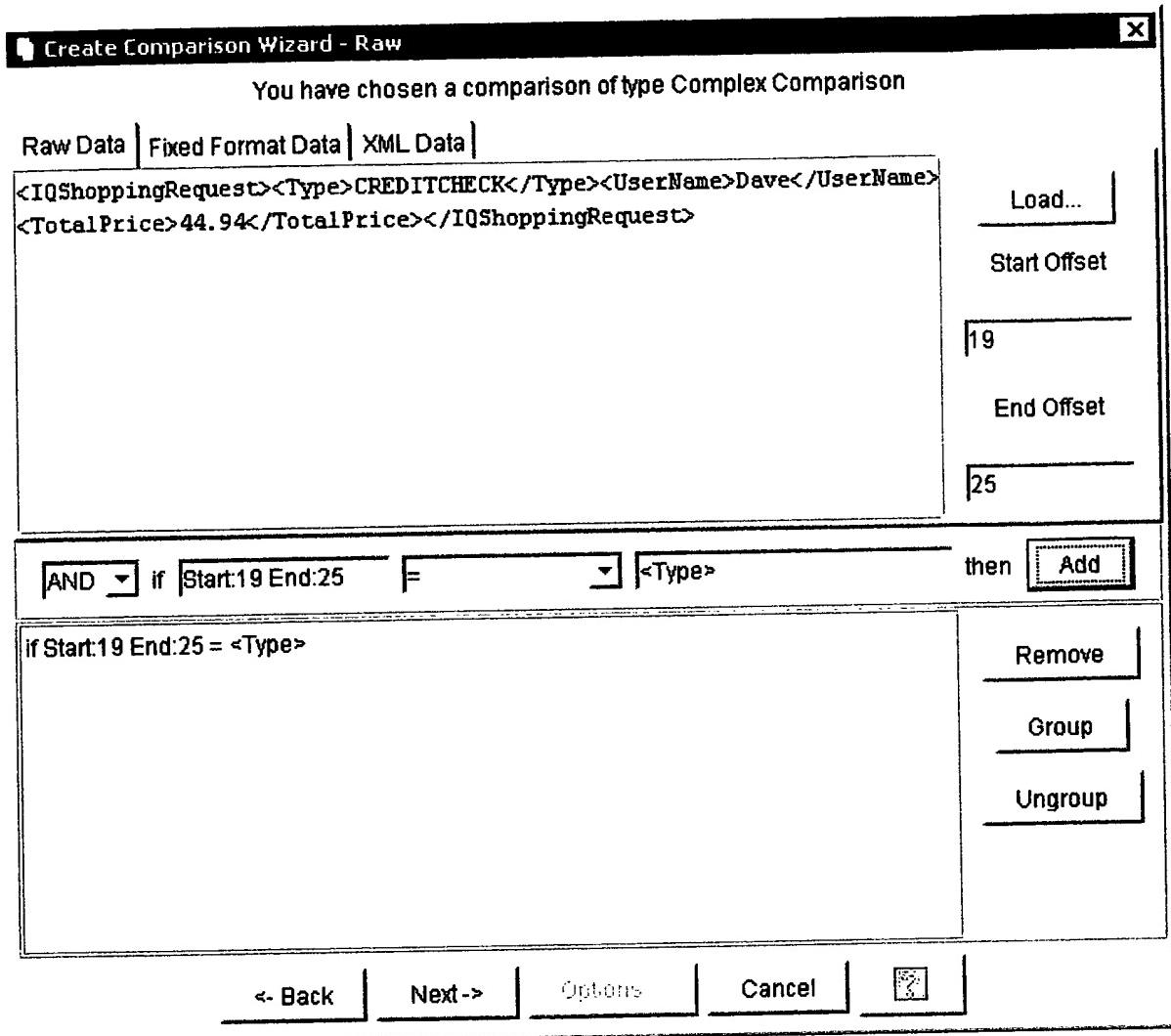


Fig 15a

```

- <Comparison baseClass="com.classiq.application.simulator.BaseComparison" className="Raw">
  <!-- This is the name of the comparison (may be different than the class name) -->
  <name>Raw</name>
  <!-- The delay used by the comparison -->
  <delay>0</delay>
  <!-- The remove after first use attribute -->
  <removeUsed>false</removeUsed>
- <!--
  This section describes the compare method. It contains 1 attribute:
  1) name - This is the name of the comparison type used. This is the label one will
  see in the interface.

  -->
- <comparisonType name="Complex Comparison">
  <!-- These are fields used by the comparison, if any -->
  - <fields>
    - <templates>
      <template name="customer" />
    </templates>
  </fields>
  <!-- This is any information used for the compareInit method. Anything under this element
  can be unique for each type of comparison -->
  <init />
  <!-- This is any information used for the compare method. Anything under this element can
  be unique for each type of comparison -->
- <method>
  - <ConditionData endParenIndex="0" leadingParens="" startParenIndex="0" theAddType="AND">
    theCompareType="=" trailingParens=""
    - <theCompareValue>
      <![CDATA[ Type>CREDITCHECK</Ty ]]>
    </theCompareValue>
    - <theSelectionData>
      - <item>
        - <key>
          <![CDATA[ SELECTED_VALUE ]]>
        </key>
        - <value>
          <![CDATA[ Type>CREDITCHECK</Ty ]]>
        </value>
      </item>
      - <item>
        - <key>
          <![CDATA[ END_OFFSET ]]>
        </key>
        - <value>
          <![CDATA[ 40 ]]>
        </value>
      </item>
      - <item>
        - <key>
          <![CDATA[ TAG_NAME ]]>
        </key>
        - <value>
          <![CDATA[ last ]]>
        </value>
      </item>
      - <item>
        - <key>
          <![CDATA[ VIEWER_TYPE ]]>
        </key>
        - <value>
          <![CDATA[ 2 ]]>
        </value>
      </item>
      - <item>
        - <key>
```

FG 15 b1

```

<![CDATA[ TEMPLATE_NAME  ]]>
</key>
- <value>
  <![CDATA[ customer  ]]>
</value>
</item>
- <item>
- <key>
  <![CDATA[ START_OFFSET  ]]>
</key>
- <value>
  <![CDATA[ 20  ]]>
</value>
</item>
</theSelectionData>
</ConditionData>
</method>
</comparisonType>
- <!--
   This section describes the reply method. It contains 1 attribute:
   1) name - This is the name of the reply type used. This is the label one would :
-->
- <replyType name="Complex Reply">
  <!-- These are fields used by the reply, if any -->
- <fields>
  <templates />
</fields>
<!-- This is any information used for the replyInit method. Anything under this element can
be unique for each type of reply -->
- <init>
  <dir>file:/C:/Solstice/Integra/Enterprise/ApplicationSimulations/Comparisons/</dir>
  <file>Raw_out.txt</file>
</init>
<!-- This is any information used for the reply method. Anything under this element can be
unique for each type of reply -->
- <method>
  - <BaseReplyMessage>
    <![CDATA[ Select or enter a message  ]]>
  </BaseReplyMessage>
  <Mappings />
  <DataSubs />
</method>
</replyType>
</Comparison>

```

FIG 1562

```

public class Raw extends
com.classiq.application.simulator.BaseComparison {
private long delay;
private boolean removeUsed;
//comparison variables
public static String[] compareTemplateNames = new String[]{"customer"} ;
//reply variables
private String strFileContent;
public static String[] replyTemplateNames = new String[]{} ;
private void comparisonInit() throws Exception {

}
private void replyInit() throws Exception {
java.net.URL url = new java.net.URL
(com.classiq.util.FileUtil.getComparisonResourceFileName(getName(),
"out"));
java.io.FileInputStream fis = new java.io.FileInputStream(url.getFile
());
int i;
strFileContent = "";
while((i = fis.read()) != -1) {
strFileContent += (char)i;
}
fis.close();

}
public long getDelay() { return delay; }
public void setDelay(long d) { delay = d; }
public boolean isDelaySetable() { return true; }
public boolean getRemoveUsed() { return removeUsed; }
public void setRemoveUsed(boolean b) { removeUsed = b; }
public boolean compareMessage(String incomingMessage) throws Exception {
if (ComplexComparisonUtil.compare
(com.classiq.message.parsers.parserUtil.getValueUsingTemplate
("customer","last",incomingMessage), "=", "Type>CREDITCHECK</Ty"))
){
    return true;
}
    return false;
}
public String getReply(String incomingMessage) throws Exception {
String theReply = strFileContent;

return theReply;

}
public Raw() throws Exception {
delay = 0;
removeUsed = false;
setName("Raw");
comparisonInit();
replyInit();
}
}

```

Fig 15c

Create Comparison Wizard - Raw

You have chosen a comparison of type Complex Comparison

Raw Data Fixed Format Data XML Data

customer

Template Management ...

Field Name	Value
name (S:0 E:20)	<IQShoppingRequest><
last (S:20 E:40)	Type>CREDITCHECK</Ty

AND if last = Type>CREDITCHECK</Ty then Add

if last = Type>CREDITCHECK</Ty

Remove
Group
Ungroup

<- Back Next -> Options Cancel

F6 16c

```

- <Comparison baseClass="com.classiq.application.simulator.BaseComparison" className="Fixed">
  <!-- This is the name of the comparison (may be different than the class name) -->
  <name>Fixed</name>
  <!-- The delay used by the comparison -->
  <delay>0</delay>
  <!-- The remove after first use attribute -->
  <removeUsed>false</removeUsed>
- <!--
  This section describes the compare method. It contains 1 attribute:
  1) name - This is the name of the comparison type used. This is the label one
-->
- <comparisonType name="Complex Comparison">
  <!-- These are fields used by the comparison, if any -->
  - <fields>
    - <templates>
      <template name="customer" />
    </templates>
  </fields>
  <!-- This is any information used for the compareInit method. Anything under this element
  can be unique for each type of comparison -->
  <init />
  <!-- This is any information used for the compare method. Anything under this element can
  be unique for each type of comparison -->
  - <method>
    - <ConditionData endParenIndex="0" leadingParens="" startParenIndex="0" theAddType="AND">
      theCompareType="=" trailingParens="">
      - <theCompareValue>
        <![CDATA[ <IQShoppingRequest>< ]]>
      </theCompareValue>
      - <theSelectionData>
        - <item>
          - <key>
            <![CDATA[ SELECTED_VALUE ]]>
          </key>
          - <value>
            <![CDATA[ <IQShoppingRequest>< ]]>
          </value>
        </item>
        - <item>
          - <key>
            <![CDATA[ END_OFFSET ]]>
          </key>
          - <value>
            <![CDATA[ 20 ]]>
          </value>
        </item>
        - <item>
          - <key>
            <![CDATA[ TAG_NAME ]]>
          </key>
          - <value>
            <![CDATA[ name ]]>
          </value>
        </item>
        - <item>
          - <key>
            <![CDATA[ VIEWER_TYPE ]]>
          </key>
          - <value>
            <![CDATA[ 2 ]]>
          </value>
        </item>
        - <item>
          - <key>

```

FG 16 b1

```

        <![CDATA[ TEMPLATE_NAME  ]]>
    </key>
- <value>
    <![CDATA[ customer  ]]>
    </value>
</item>
- <item>
- <key>
    <![CDATA[ START_OFFSET  ]]>
    </key>
- <value>
    <![CDATA[ 0  ]]>
    </value>
</item>
</theSelectionData>
</ConditionData>
</method>
</comparisonType>
- <!--
    This section describes the reply method. It contains 1 attribute:
    1) name - This is the label one would :
-->
- <replyType name="Complex Reply">
    <!-- These are fields used by the reply, if any -->
- <fields>
    <templates />
</fields>
<!-- This is any information used for the replyInit method. Anything under this element can
be unique for each type of reply -->
- <init>
    <dir>file:/C:/Solstice/Integra/Enterprise/ApplicationSimulations/Comparisons/</dir>
    <file>Fixed_out.txt</file>
</init>
<!-- This is any information used for the reply method. Anything under this element can be
unique for each type of reply -->
- <method>
    - <BaseReplyMessage>
        <![CDATA[ Select or enter a message  ]]>
    </BaseReplyMessage>
    <Mappings />
    <DataSubs />
</method>
</replyType>
</Comparison>

```

fg 1662

```

public class Fixed extends
com.classiq.application.simulator.BaseComparison {
private long delay;
private boolean removeUsed;
//comparison variables
public static String[] compareTemplateName = new String[]{"customer"} ;
//reply variables
private String strFileContent;
public static String[] replyTemplateName = new String[]{};
private void comparisonInit() throws Exception {

}
private void replyInit() throws Exception {
java.net.URL url = new java.net.URL
(com.classiq.util.FileUtil.getComparisonResourceFileName(getName(),
"out"));
java.io.FileInputStream fis = new java.io.FileInputStream(url.getFile
());
int i;
strFileContent = "";
while((i = fis.read()) != -1) {
strFileContent += (char)i;
}
fis.close();
}
public long getDelay() { return delay; }
public void setDelay(long d) { delay = d; }
public boolean isDelaySetable() { return true; }
public boolean getRemoveUsed() { return removeUsed; }
public void setRemoveUsed(boolean b) { removeUsed = b; }
public boolean compareMessage(String incomingMessage) throws Exception {
if (ComplexComparisonUtil.compare
(com.classiq.message.parsers.parserUtil.getValueUsingTemplate
("customer", "name", incomingMessage), "=", "<IQShoppingRequest><"))
){
    return true;
}
    return false;
}
public String getReply(String incomingMessage) throws Exception {
String theReply = strFileContent;

return theReply;
}
public Fixed() throws Exception {
delay = 0;
removeUsed = false;
setName("Fixed");
comparisonInit();
replyInit();
}
}

```

fg 16c

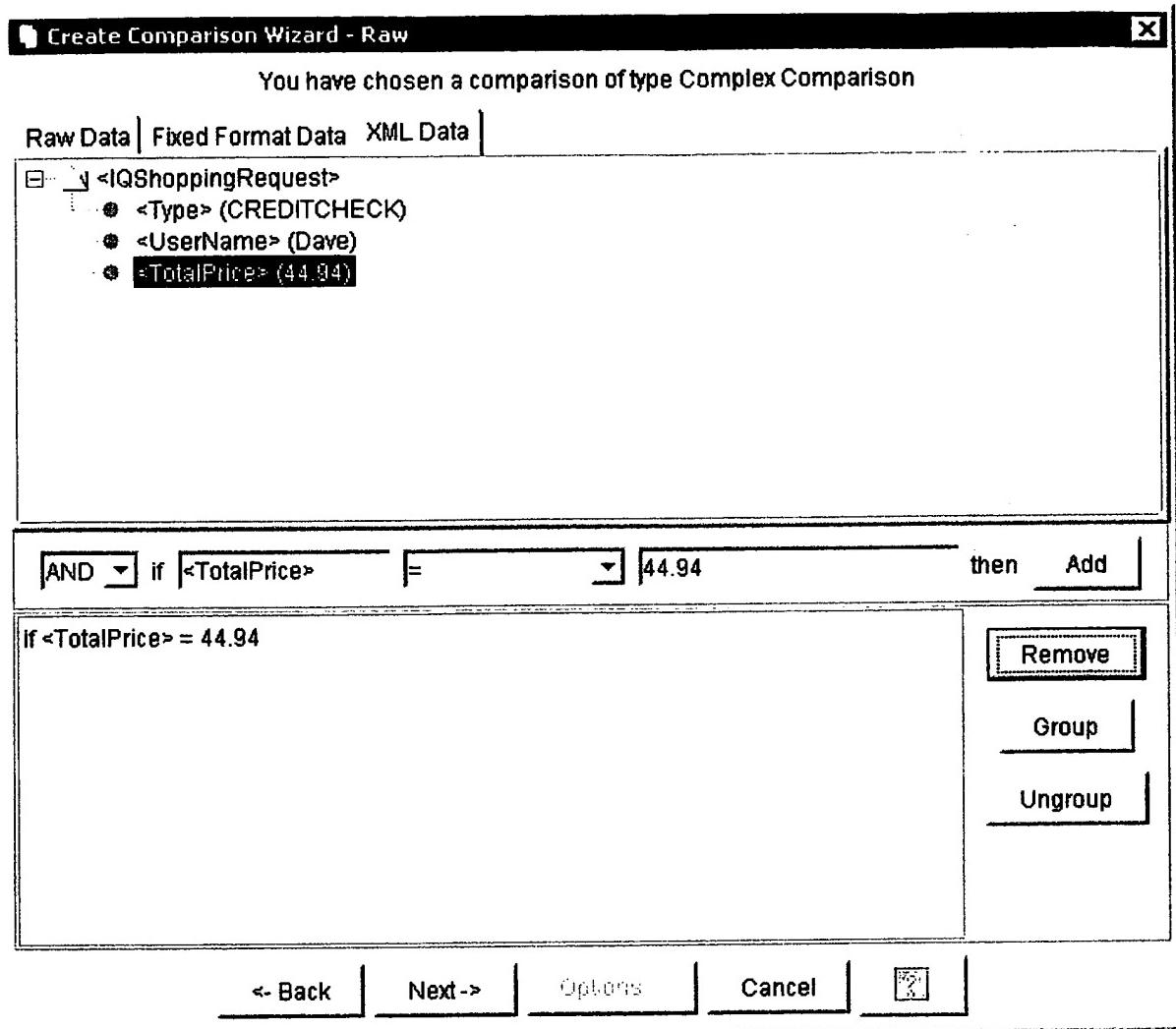


FIG 17a

```

- <Comparison baseClass="com.classiq.application.simulator.BaseComparison" className="XML">
  <!-- This is the name of the comparison (may be different than the class name) -->
  <name>XML</name>
  <!-- The delay used by the comparison -->
  <delay>0</delay>
  <!-- The remove after first use attribute -->
  <removeUsed>false</removeUsed>
- <!--
  This section describes the compare method. It contains 1 attribute:
  1) name - This is the label and u
-->
- <comparisonType name="Complex Comparison">
  <!-- These are fields used by the comparison, if any -->
  - <fields>
    <templates />
  </fields>
  <!-- This is any information used for the compareInit method. Anything under this element
  can be unique for each type of comparison -->
  <init />
  <!-- This is any information used for the compare method. Anything under this element can
  be unique for each type of comparison -->
- <method>
  - <ConditionData endParenIndex="0" leadingParens="" startParenIndex="0" theAddType="AND">
    theCompareType="=" trailingParens=""
  - <theCompareValue>
    <![CDATA[ 44.94 ]]>
  </theCompareValue>
  - <theSelectionData>
    - <item>
      - <key>
        <![CDATA[ SELECTED_VALUE ]]>
      </key>
      - <value>
        <![CDATA[ 44.94 ]]>
      </value>
    </item>
    - <item>
      - <key>
        <![CDATA[ XML_PATH ]]>
      </key>
      - <value>
        <![CDATA[ [TotalPrice*0, IQShoppingRequest*0] ]]>
      </value>
    </item>
    - <item>
      - <key>
        <![CDATA[ TAG_NAME ]]>
      </key>
      - <value>
        <![CDATA[ <TotalPrice> ]]>
      </value>
    </item>
    - <item>
      - <key>
        <![CDATA[ VIEWER_TYPE ]]>
      </key>
      - <value>
        <![CDATA[ 3 ]]>
      </value>
    </item>
    - <item>
      - <key>
        <![CDATA[ XML_PATH_STRING ]]>
      </key>
    </item>
  </theSelectionData>

```

56 176'

```

- <value>
  <![CDATA[ [TotalPrice*0, IQShoppingRequest*0] ]]>
</value>
</item>
</theSelectionData>
</ConditionData>
</method>
</comparisonType>
- <!--
  this section describes the reply method. It contains 1 attribute:
    1) name - This is the name of the reply type used. This is the label one would
-->
- <replyType name="Complex Reply">
  <!-- These are fields used by the reply, if any -->
  <fields>
    <templates />
  </fields>
  <!-- This is any information used for the replyInit method. Anything under this element can
  be unique for each type of reply -->
  <!-->
  <init>
    <dir>file:/C:/Solstice/Integra/Enterprise/ApplicationSimulations/Comparisons/</dir>
    <file>XML_out.txt</file>
  </init>
  <!-- This is any information used for the reply method. Anything under this element can be
  be unique for each type of reply -->
  <!-->
  <method>
    <BaseReplyMessage>
      <![CDATA[ Select or enter a message ]]>
    </BaseReplyMessage>
    <Mappings />
    <DataSubs />
  </method>
</replyType>
</Comparison>

```

fg 1762

```

public class XML extends
com.classiq.application.simulator.BaseComparison {
private long delay;
private boolean removeUsed;
//comparison variables
public static String[] compareTemplateName = new String[]{} ;
//reply variables
private String strFileContent;
public static String[] replyTemplateName = new String[]{} ;
private void comparisonInit() throws Exception {
}
private void replyInit() throws Exception {
java.net.URL url = new java.net.URL
(com.classiq.util.FileUtil.getComparisonResourceFileName(getName(),
"out"));
java.io.FileInputStream fis = new java.io.FileInputStream(url.getFile
());
int i;
strFileContent = "";
while((i = fis.read()) != -1) {
strFileContent += (char)i;
}
fis.close();
}
public long getDelay() { return delay; }
public void setDelay(long d) { delay = d; }
public boolean isDelaySetable() { return true; }
public boolean getRemoveUsed() { return removeUsed; }
public void setRemoveUsed(boolean b) { removeUsed = b; }
public boolean compareMessage(String incomingMessage) throws Exception {
if (ComplexComparisonUtil.compare
(com.classiq.message.parsers.parserUtil.getValue(incomingMessage,new
String[]{"TotalPrice*0","IQShoppingRequest*0",}), "=", "44.94")
){
return true;
}
return false;
}
public String getReply(String incomingMessage) throws Exception {
String theReply = strFileContent;
return theReply;
}
public XML() throws Exception {
delay = 0;
removeUsed = false;
setName("XML");
comparisonInit();
replyInit();
}
}

```

Fig 17c

FIG 18

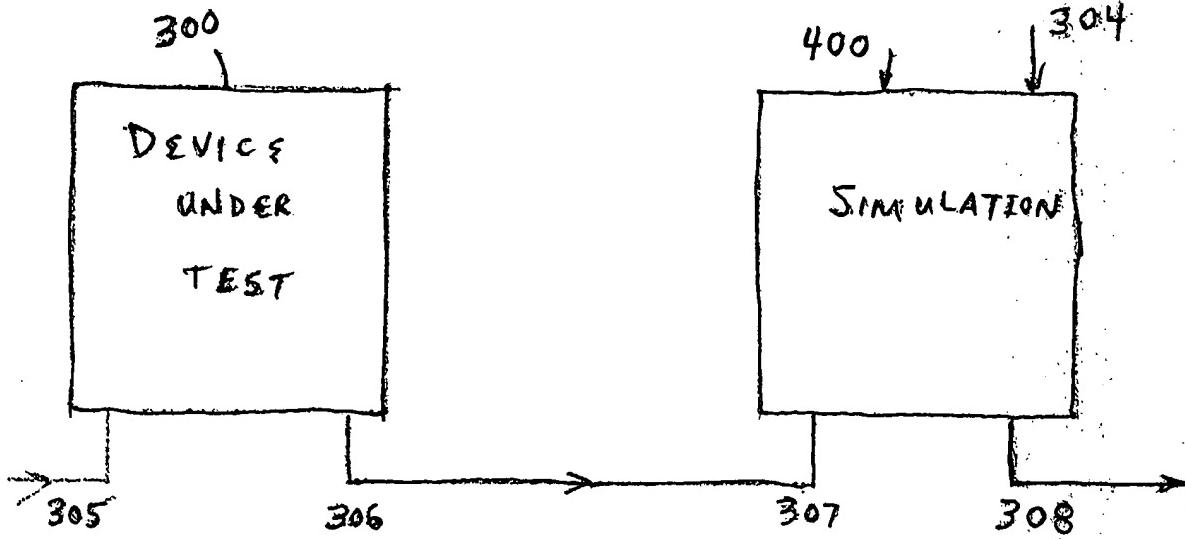


FIG 19

